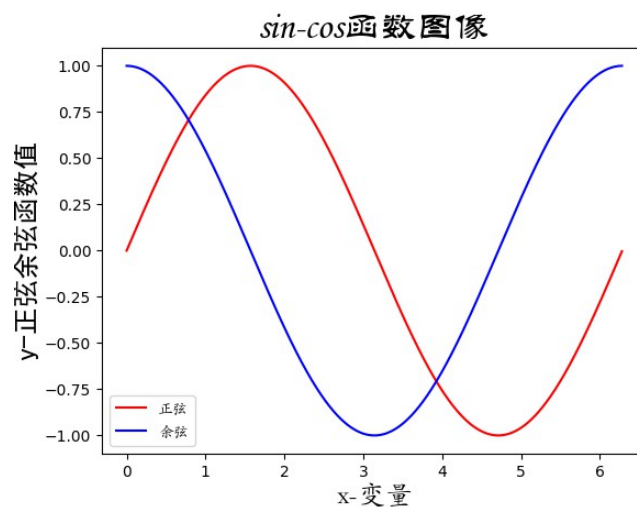


Python 程序设计教案

本次授课内容	15.1 matplotlib 简介 15.2 绘制带有中文标题、标签和图例的图 15.3 绘制散点图 15.4 绘制饼状图 15.5 绘制柱状图	
本次课的教学目的	了解 matplotlib 基本用法 掌握折线图、散点图、饼状图、柱状图的绘制	
本次课教学重点与难点	折线图、散点图、饼状图、柱状图的绘制	
教学方法 教学手段	PPT、边讲边练	
课堂教学 时间分配	教学内容	时间分配 (分)
课堂教学设计	首先介绍 matplotlib 的安装和简单使用，然后通过案例讲解折线图、散点图、饼状图、柱状图的绘制。	
实 验	教材 15.5 节之前的所有例题。	
思考题及作业题	本章所有课后习题。	
备 注		
教学后记		

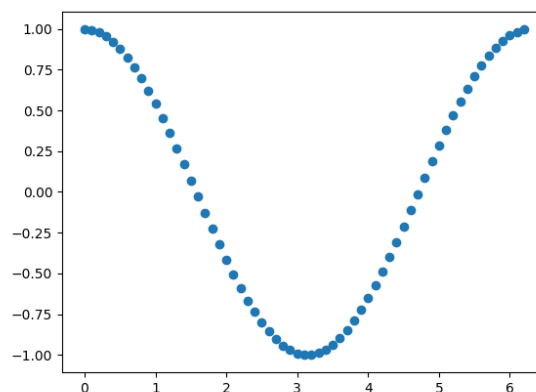
课 堂 重 点 内 容	第 一 节 课
	例 15-1 绘制带有中文标题、标签和图例的正弦和余弦图像。 <pre>import numpy as np import matplotlib.pyplot as plt import matplotlib.font_manager as fm t = np.arange(0.0, 2.0*np.pi, 0.01) #自变量取值范围 s = np.sin(t) #计算正弦函数值 z = np.cos(t) #计算余弦函数值 plt.plot(t, #x 轴坐标 s, #y 轴坐标 label='正弦', #标签 color='red') #颜色 plt.plot(t, z, label='余弦', color='blue') plt.xlabel('x-变量', #标签文本 fontproperties='STKAITI', #字体 fontsize=18) #字号 plt.ylabel('y-正弦余弦函数值', fontproperties='simhei', fontsize=18) plt.title('sin-cos 函数图像', #标题文本 fontproperties='STLITI', #字体 fontsize=24) #字号 myfont = fm.FontProperties(fname=r'C:\Windows\Fonts\STKAITI.ttf') #创建字体对象 plt.legend(prop=myfont) #显示图例 plt.show() #显示绘制的结果图像</pre>



例 15-2 绘制余弦曲线散点图。

```
import numpy as np
import matplotlib.pyplot as plt

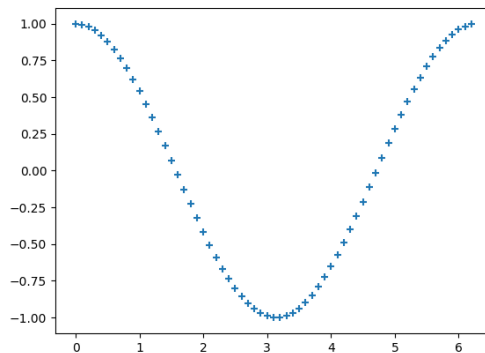
x = np.arange(0, 2.0*np.pi, 0.1)  #x 轴数据
y = np.cos(x)                      #y 轴数据
plt.scatter(x, y)                  #绘制散点图
plt.show()                         #显示绘制的结果图像
```



例 15-3 设置散点图的线宽、散点符号及大小。

```
import numpy as np
import matplotlib.pyplot as plt

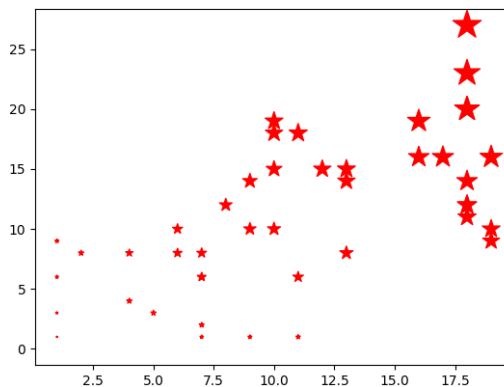
x = np.arange(0, 2.0*np.pi, 0.1)  #x 轴数据
y = np.cos(x)                      #y 轴数据
plt.scatter(x, y, s=40, linewidths=6, marker='+')
plt.show()
```



例 15-4 绘制大小与位置有关的红色散点五角星。

```
import matplotlib.pyplot as plt
from numpy.random import randint

x = randint(1, 20, 50)      #模拟 x 轴数据
y = x + randint(-10, 10, 50) #生成 y 轴数据
plt.scatter(x, y, s=x*y, c='r', marker='*')
plt.show()
```



第 二 节 课

例 15-5 饼状图绘制与属性设置。

```
import numpy as np
import matplotlib.pyplot as plt

labels = ('Frogs', 'Hogs', 'Dogs', 'Logs')
colors = ('#FF0000', 'yellowgreen', 'gold', 'blue')
explode = (0, 0.02, 0, 0.08)      #使所有饼状图中第 2 片和第 4 片裂开

fig = plt.figure(num=1,figsize=(10,8),dpi=110,facecolor='white')
```

```

ax = fig.gca()                                #获取当前轴
ax.pie(np.random.random(4), explode=explode, labels=labels,
       colors=colors, pctdistance=0.8, autopct='%1.1f%%',
       shadow=True, startangle=90, radius=0.25, center=(0, 0),
       counterclock=False, frame=True)
ax.pie(np.random.random(4), explode=explode, labels=labels,
       colors=colors, autopct='%1.1f%%', shadow=True,
       startangle=45, radius=0.25, center=(1, 1), frame=True)
ax.pie(np.random.random(4), explode=explode, labels=labels,
       colors=colors, autopct='%1.1f%%', shadow=True,
       startangle=90, radius=0.25, center=(0, 1), frame=True)
ax.pie(np.random.random(4), explode=explode, labels=labels,
       colors=colors, autopct='%1.2f%%', shadow=False,
       startangle=135, radius=0.35, center=(1, 0), frame=True)

ax.set_xticks([0, 1])                        #设置 x 坐标轴刻度
ax.set_yticks([0, 1])                       #设置 y 坐标轴刻度

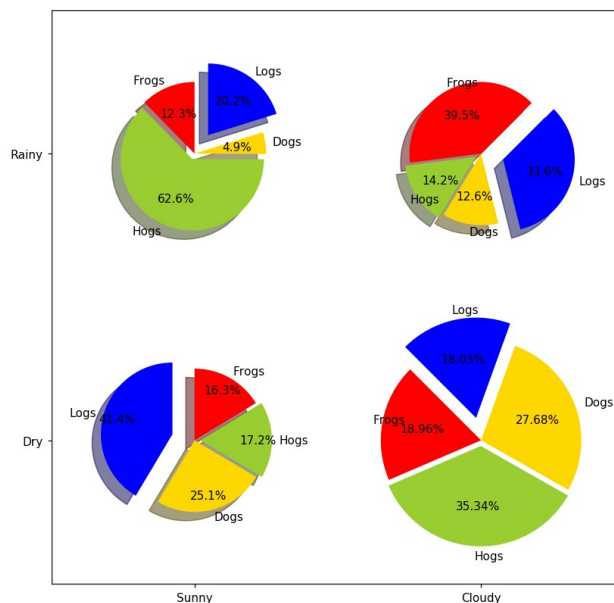
ax.set_xticklabels(["Sunny", "Cloudy"])#设置坐标轴刻度上的标签
ax.set_yticklabels(["Dry", "Rainy"])

ax.set_xlim((-0.5, 1.5))                    #设置坐标轴跨度
ax.set_ylim((-0.5, 1.5))

ax.set_aspect('equal')                      #设置纵横比相等

plt.show()

```



例 15-6 绘制柱状图并设置图形属性和文本标注。

```
import numpy as np
```

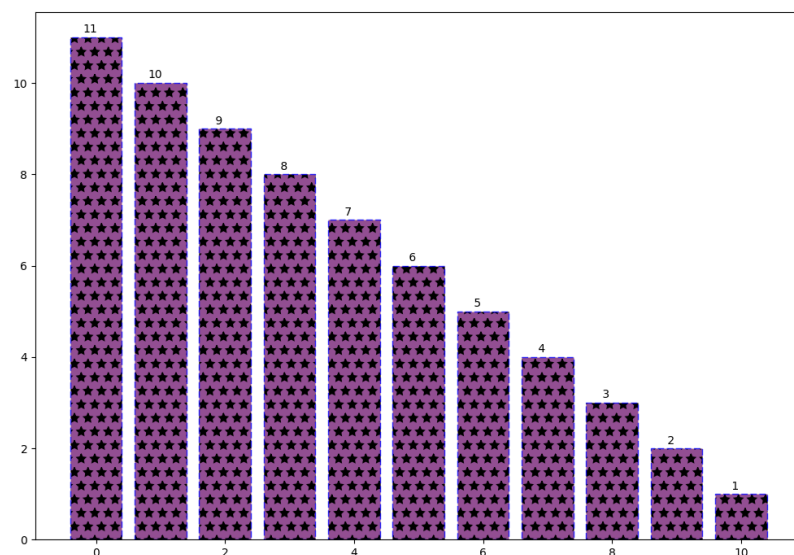
```
import matplotlib.pyplot as plt

#生成测试数据
x = np.linspace(0, 10, 11)
y = 11-x

#绘制柱状图
plt.bar(x, y, color='#772277', alpha=0.8, edgecolor='blue',
        linestyle='--', linewidth=1, hatch='*')

#为每个柱形添加文本标注
for xx, yy in zip(x,y):
    plt.text(xx-0.2, yy+0.1, '%2d' % yy)

#显示图形
plt.show()
```

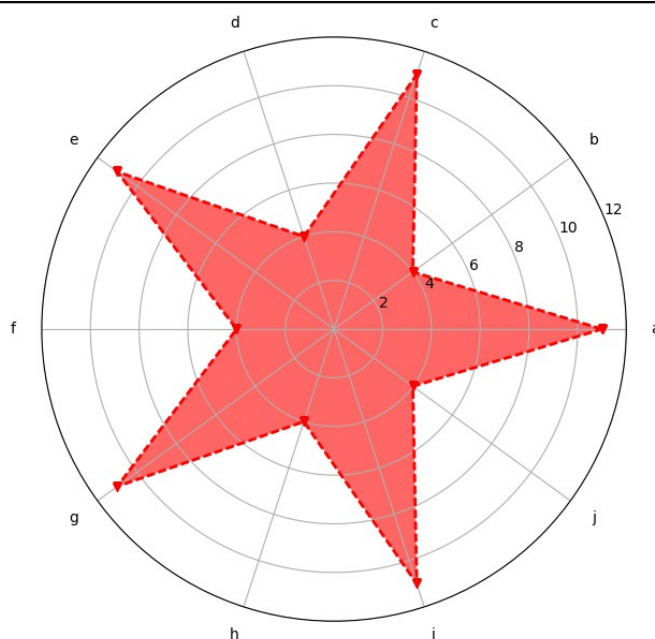


Python 程序设计教案

本次授课内容	15.6 绘制雷达图 15.7 绘制三维图形 15.8 切分绘图区域 15.9 设置图例 15.10 设置坐标轴刻度距离和文本	
本次课的教学目的	掌握雷达图、三维图的绘制 理解绘图区域的切分原理 掌握图例的设置方法 掌握设置坐标轴刻度距离和文本的方法	
本次课教学重点与难点	雷达图、三维图的绘制 绘图区域的切分原理	
教学方法 教学手段	PPT、边讲边练	
课堂教学 时间分配	教学内容	时间分配 (分)
课堂教学设计	通过实际案例演示雷达图、三维图形的绘制，演示绘图区域切分、设置图例、设置坐标轴刻度距离和文本的方法。	
实 验	教材 15.6 至 15.10 节的所有例题。	
思考题及作业题	本章所有课后习题。	
备 注		
教学后记		

课 堂 重 点 内 容	第 一 节 课
	例 15-7 绘制雷达图。 <pre>import numpy as np import matplotlib.pyplot as plt labels = np.array(list('abcdefghij')) #设置标签 data = np.array([11,4]*5) #创建模拟数据 dataLength = len(labels) #数据长度 #angles 数组把圆周等分为 dataLength 份 angles = np.linspace(0, 2*np.pi, dataLength, endpoint=False) data = np.append(data, data[0]) angles = np.append(angles, angles[0]) #首尾相接，使得曲线闭合 #绘制雷达图 plt.polar(angles, data, 'rv--', linewidth=2) #设置角度网格标签 plt.thetagrids(angles*180/np.pi, labels) #设置填充色 plt.fill(angles, data, facecolor='r', alpha=0.6) plt.ylim(0,12) #设置坐标跨度 plt.show() #显示绘图结果</pre>

详
解



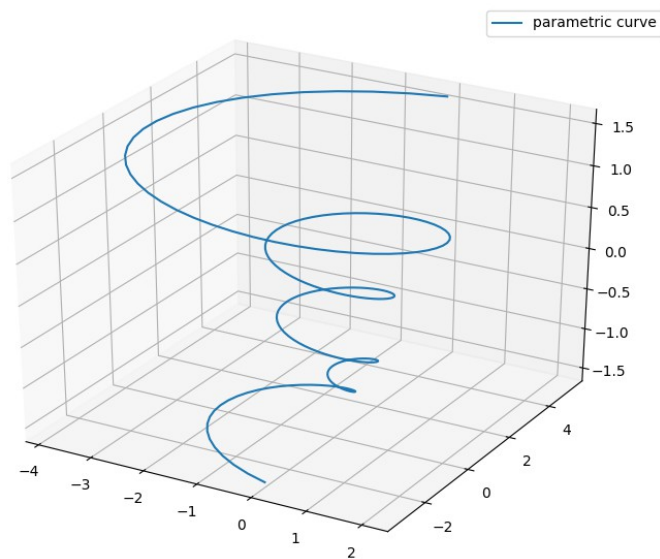
例 15-8 绘制三维曲线。

```
import matplotlib as mpl
from mpl_toolkits.mplot3d import Axes3D
import numpy as np
import matplotlib.pyplot as plt

mpl.rcParams['legend.fontsize'] = 10    #设置图例字号
fig = plt.figure()
ax = fig.gca(projection='3d')           #绘制三维图形
theta = np.linspace(-4 * np.pi, 4 * np.pi, 200)
z = np.linspace(-4, 4, 200)*0.4        #创建模拟数据
                                         #z 的步长应与 theta 一致

r = z**3 + 1
x = r * np.sin(theta)
y = r * np.cos(theta)
ax.plot(x, y, z, label='parametric curve')
ax.legend()                             #显示图例

plt.show()                              #显示绘制结果
```

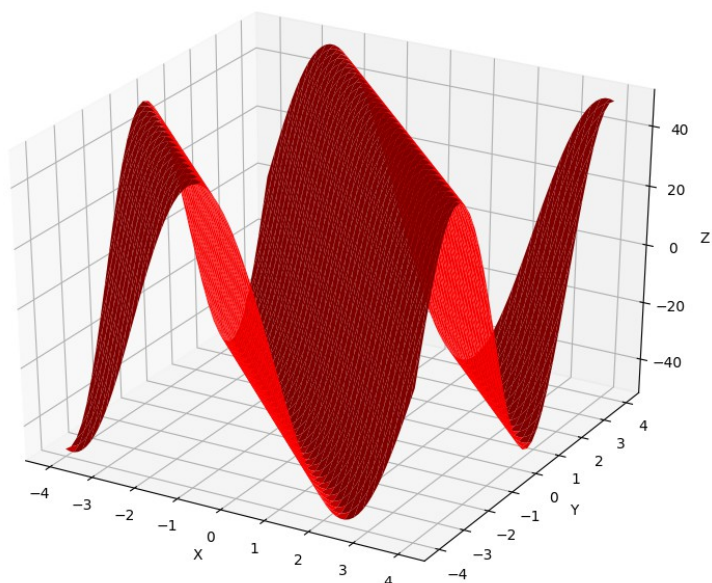


例 15-9 绘制三维曲面。

```
import numpy as np
import matplotlib.pyplot as plt
import mpl_toolkits.mplot3d

x,y = np.mgrid[-4:4:80j, -4:4:40j]
z = 50 * np.sin(x+y)                    #创建测试数据
ax = plt.subplot(projection='3d')        #绘制三维图形
ax.plot_surface(x, y, z, rstride=2, cstride=1, color='red', )
ax.set_xlabel('X')                      #设置坐标轴标签
ax.set_ylabel('Y')
ax.set_zlabel('Z')

plt.show()
```



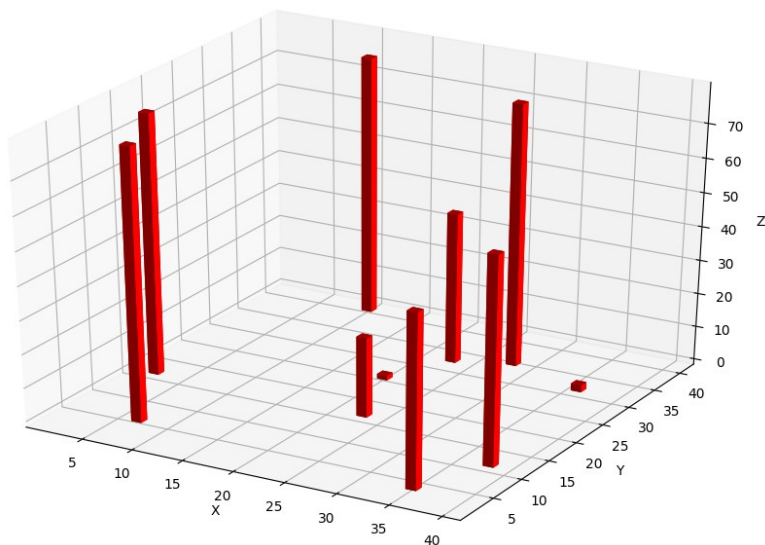
例 15-10 绘制三维柱状图。

```
import numpy as np
import matplotlib.pyplot as plt
import mpl_toolkits.mplot3d

x = np.random.randint(0, 40, 10)      #创建测试数据
y = np.random.randint(0, 40, 10)
z = 80*abs(np.sin(x+y))
ax = plt.subplot(projection='3d')     #绘制三维图形

ax.bar3d(x, y, np.zeros_like(z), dx=1, dy=1, dz=z, color='red')
#设置面片颜色为红色
ax.set_xlabel('X')                   #设置坐标轴标签
ax.set_ylabel('Y')
ax.set_zlabel('Z')

plt.show()
```



第 二 节 课

例 15-11 切分绘图区域并绘制图形。

```
import numpy as np
import matplotlib.pyplot as plt

x= np.linspace(0, 2*np.pi, 500)      #创建自变量数组
y1 = np.sin(x)                        #创建函数值数组
y2 = np.cos(x)
y3 = np.sin(x*x)
```

```
plt.figure()                                #创建图形

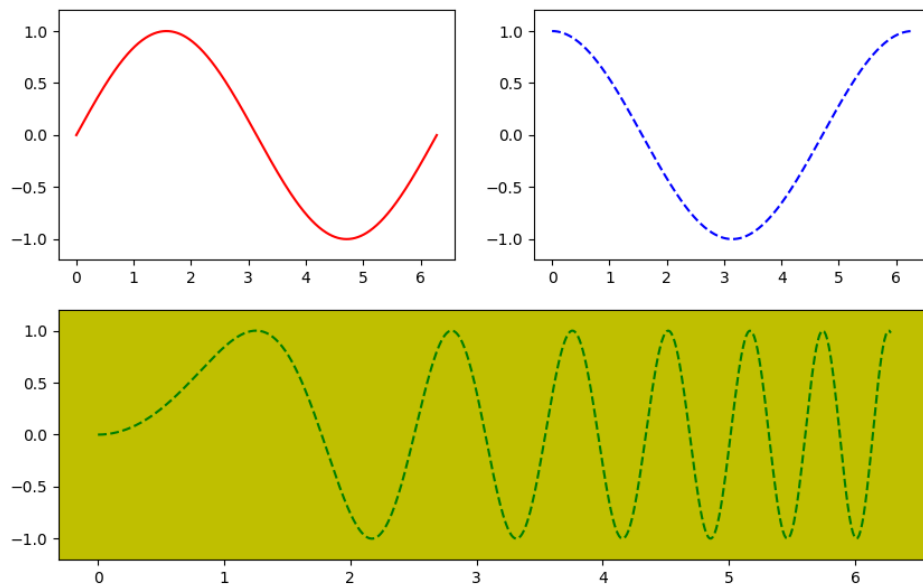
ax1 = plt.subplot(2, 2, 1)
ax2 = plt.subplot(2,2,2)                    #选择两行两列的第二个区域
ax3 = plt.subplot(212, facecolor='y')

plt.sca(ax1)                                #选择 ax1
plt.plot(x, y1, color='red')                #绘制红色曲线
plt.ylim(-1.2, 1.2)                         #限制 y 坐标轴范围

plt.sca(ax2)                                #选择 ax2
plt.plot(x, y2, 'b--')                      #绘制蓝色虚线
plt.ylim(-1.2,1.2)

plt.sca(ax3)                                #选择 ax3
plt.plot(x, y3, 'g--')
plt.ylim(-1.2, 1.2)

plt.show()
```



例 15-12 设置图例显示公式。

```
import numpy as np
import matplotlib.pyplot as plt

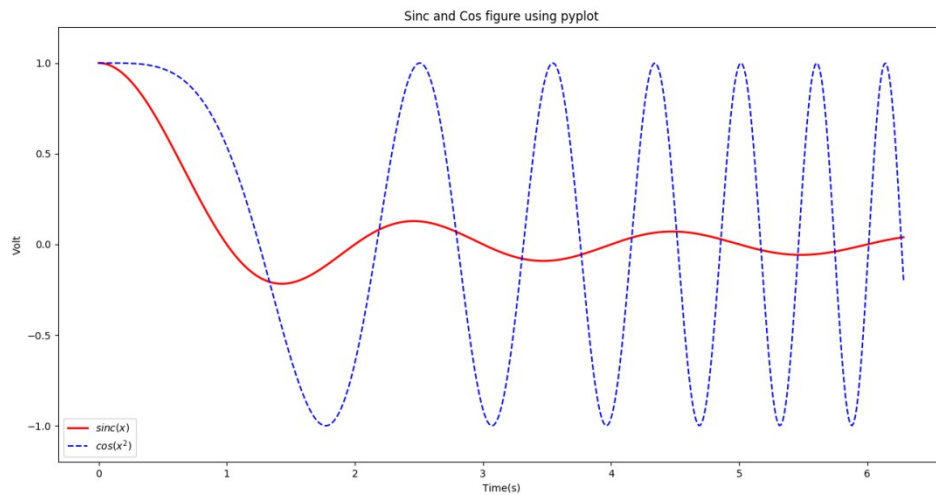
x = np.linspace(0, 2*np.pi, 500)
y = np.sinc(x)
z = np.cos(x*x)
plt.figure(figsize=(8,4))

plt.plot(x, y, label='$\text{sinc}(x)$', color='red', linewidth=2)
```

```
plt.plot(x, z, 'b--', label='$\cos(x^2)$')

plt.xlabel('Time(s)')
plt.ylabel('Volt')
plt.title('Sinc and Cos figure using pyplot')
plt.ylim(-1.2,1.2)
plt.legend()                #显示图例

plt.show()                  #显示绘图结果
```



例 15-13 设置图例位置、背景颜色、边框颜色等属性。

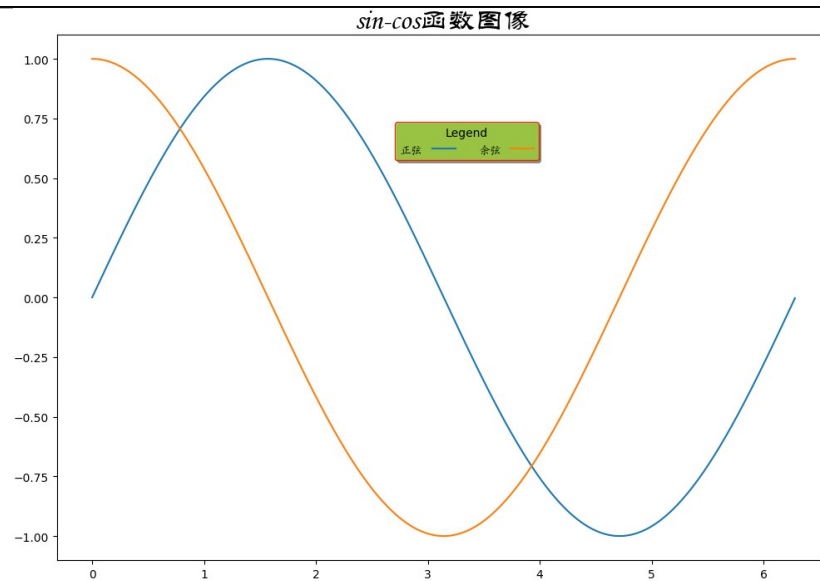
```
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.font_manager as fm

t = np.arange(0.0, 2*np.pi, 0.01)
s = np.sin(t)
z = np.cos(t)

plt.plot(t, s, label='正弦')
plt.plot(t, z, label='余弦')
plt.title('sin-cos 函数图像', fontproperties='STLITI', fontsize=24)

myfont = fm.FontProperties(fname=r'C:\Windows\Fonts\STKAITI.ttf')
plt.legend(prop=myfont, title='Legend', loc='lower left',
           bbox_to_anchor=(0.43,0.75), shadow=True,
           facecolor='yellowgreen', edgecolor='red',
           ncol=2, markerfirst=False)

plt.show()
```



例 15-14 设置坐标轴刻度距离和文本。

```
import numpy as np
import matplotlib.pyplot as plt

x = np.arange(0, 2*np.pi, 0.01)
y = np.sin(x)
plt.plot(x, y)

plt.xticks(np.arange(0, 2*np.pi, 0.5))
plt.yticks([-1, -0.5, 0, 0.75, 1],
            ['负一', '负零点五', '零', '零点七五', '一'],
            fontproperties='STKAITI')

plt.show()
```

